REMARKS

The claims are claims 1, 5, 67, 10, 11, 12, 16, 17, 19, 20, 21 and 22.

Claims 1, 5, 7, 10, 11, 12, 16, 17, 19, 20, 21 and 22 are amended. Claims 2, 3, 4, 6, 8, 9, 13, 14, 15 and 19 are canceled. Claims 1, 5, 7, 10, 11, 12, 16, 17, 19, 20, 21 and 22 are amended to distinguish over the new references cited in the new rejections in the FINAL REJECTION.

Claims 1, 5, 10, 11, 12, and 16 were rejected under 35 U.S.C. 102(b) as anticipated by Yung U.S. Patent Number 5,809,324.

Claim 1 recites subject matter not anticipated by Yung. Claim 1 recites "an annul word memory for storing an annul code having a plurality of annul bits, each annul bit having a one-to-one correspondence to one instruction of a group of instructions in the pipeline." This recitation of the annul word memory distinguishes over FIFO 66 disclosed in Yung. Yung states at column 4, lines 9 to 18:

> "At the top of FIG. 3, an outstanding instruction FIFO 66 is illustrated. As shown, the FIFO has 7 rows corresponding to 7 stages or cycles of the pipeline, with corresponding instructions being stored in 4 positions in each row, extending from 1 through 28 as shown. Each row thus stores an instruction group. The instructions corresponding to the positions in the FIFO will be found in the pipeline at the same cycle in the functional units. Instruction FIFO control logic 71 controls the operation of outstanding instruction FIFO 66."

This disclosure makes clear that FIFO 66 of Yung has one bit position for each dispatch slot of the 4 scalar machine. Note that Yung teaches at column 4, line 65 to column 5, line 16 as illustrated in Figure 4 numerous conditions under which less than four instructions are scheduled for a clock cycle. Under these

conditions the bit position within FIFO 66 corresponding to a non dispatched instruction is meaningless. In addition, this makes clear that bit positions within FIFO 66 do not have the one-to-one correspondence to instructions recited in claim 1. Accordingly, claim 1 is allowable over Yung.

Claim 5 recites further subject matter not anticipated by Yung. Claim 5 recites "coupling only the integer number N of the annul bits to the plurality of execution units which are scheduled to execute in the given clock cycle whereby on an immediately following clock cycle annul bits beginning at an N+1 annul bit are coupled to execution units scheduled to execute of that following cycle." Yung teaches coupling only N bits to the functional units when N is less than the maximum number of instructions that are scheduled for dispatch that clock cycle. However, Yung fails to teach that the next cycle couples the N+1 bit to a functional unit. In a case where less than the maximum number of instructions are scheduled for dispatch during a clock cycle, Yung teaches that the remaining bits in FIFO 66 in that column are not used. Thus Yung uses the first bit in the next column is the next bit used. In the case where less than the maximum number of instructions were dispatched the prior clock cycle, Yung skips over bits within FIFO 66 not used during the prior cycle. Thus Yung fails to anticipate using the N+1 annul bit under these conditions. Accordingly, claim 5 is not anticipated by Yung.

Claim 10 recites subject matter not anticipated by Yung. claim 10 recites "the annul code is generated in response to one or more constant generating instructions and loaded into the annul word memory." This language requires generating of a constant, then storing that constant in the annul word memory. Yung does not anticipate these two steps. The portions of Yung cited in the FINAL REJECTION fail to anticipate this two step process. Yung

states at the ABSTRACT, lines 3 to 6 and at column 2, lines 18 to 21:

> "A mispredicted branch is handled by setting a valid bit to invalid for instructions following the branch instruction in an outstanding instruction FIFO."

Yung states at column 3, lines 51 to 53:

> "This could be done, for example, by setting all the valid bits corresponding to these instructions in an outstanding instruction FIFO to an invalid state."

Yung states at column 5, lines 46 to 54:

> "The valid bit is then set to an invalid state for all the instructions following the branch instruction (step N). This may be accomplished by branch unit 18 of FIG. 1 sending a control signal to instruction FIFO control logic 71 that the branch was mispredicted, identifying the branch instruction. Control logic 71 will then locate the instruction information in outstanding instruction FIFO 66 and set the appropriate valid bits to an invalid state."

Yung states at column 6, lines 25 to 29:

> "In another alternate embodiment, a single valid bit could be used for each stage or cycle in outstanding instruction FIFO register 66. For a mispredicted branch target, the entire line of the outstanding instruction FIFO corresponding to that stage or cycle could be invalidated."

Each of these sections of Yung which were cited by the Examiner in the FINAL REJECTION implies change of the state of bits of FIFO directly by some control logic. The quoted portion of column 5 states that branch unit 18 transmits control signals to instruction FIFO control logic 71 to alter the state of bits within FIFO 66. None of these portions of Yung disclose generation of constant data

which is then loaded into a memory as recited in claim 10. Accordingly, claim 10 is not anticipated by Yung.

Claim 11 recites subject matter not anticipated by Yung. Claim 11 recites "the annul code is loaded into the annul word memory from a memory." The FINAL REJECTION cited the same portions of Yung against claim 11 as against claim 10. The Applicant respectfully submits that Yung fails to teach an annul code is loaded into FIFO 66 from a memory as recited in claim 11. None of the portions of Yung cited in the FINAL REJECTION teach loading FIFO 66 from a memory. Instead Yung teaches changing valid bits in FIFO 66 via instruction FIFO control logic 71 under the control of branch unit 18. This invalidation is not disclosed as resulting from loading from a memory as recited in claim 11. Accordingly, claim 11 is not anticipated by Yung.

Claim 12 recites subject matter not anticipated by Yung. Claim 12 recites "the annul code is an immediate value in an immediate operand instruction passing through the pipeline loaded into the annul word memory in response to execution of the immediate operand instruction." The FINAL REJECTION cites the same portions of Yung against claim 12 as against claims 10 and 11. However, Yung fails to disclose any immediate instructions. As known in the art, an immediate instruction includes constant data within the instruction. Claim 12 recites that this data is loaded into the annul word memory. Yung cannot make this subject matter obvious because it fails to include any mention of an immediate instruction. Accordingly, claim 12 is not anticipated by Yung.

Claim 16 recites subject matter not anticipated by Yung. Claim 16 recites the circuitry for preventing "prevents the first group of instructions from altering the architected state in response to corresponding annul bits of the annul code having the first state if the condition is not satisfied" and "prevents the second group of instructions from altering the architected state in

response to corresponding annul bits of the annul code having the first state if the condition is satisfied." Yung teaches invalidating instructions following a taken conditional branch instruction. This prevents those instructions from changing the architected state. However, if the conditional branch is not taken, Yang does not teach that the bits in FIFO 66 are changed to invalidate a different set of instructions. However, claim 16 recites that annul bits prevent a second set of instructions if the condition is not satisfied. Yang only teaches invalidating a set of instructions upon one contingency. Claim 16 recites invalidating differing sets of instructions based on the condition satisfied or not satisfied. Accordingly, claim 16 is not anticipated by Yang.

Claim 22 was rejected under 35 U.S.C. 103(a) as made obvious by the combination of Yung U.S. Patent Number 5,809,324 and Lass U.S. Patent Number 5,165,025.

Claim 22 recites subject matter not made obvious by the combination of Yung and Lass. Claim 22 recites identifying a group of instructions in a tree including plural conditional branch instructions and forming two annul codes dependent upon the two branches of a conditional branch instruction taking place "at compile time prior to execution." The Applicant respectfully submits that neither Yung nor Lass include any teaching regarding forming differing annul codes at compile time before execution. Yang teaches changing bits to invalid at execution time upon detection of a branch instruction. Thus Yung fails to teach the pre-forming recited in claim 22. Lass teaches arranging the two branch paths in alternating addresses. Lass fails to teach pre-forming an annul word as recited in claim 22. Accordingly, claim 22 is not made obvious by the combination of Yung and Lass.

Claims 7, 17, 20 and 21 were rejected under 35 U.S.C. 103(a) as made obvious by the combination of Yung U.S. Patent Number 5,809,324 and InstantWeb's Free On-line Computing Dictionary.

Claim 7 recites subject matter not made obvious by the combination of Yung and InstantWeb. Claim 7 recites "the annul code based on a relationship of the given iteration to the integer M number of iterations preventing differing instructions from altering the architected state during different iterations." While InstantWeb teaches software loops which may be formed of the conditional branch instructions of Yung, the combination fails to teach this preventing differing instructions from completing during differing iterations of the loop. The portions of Yung cited in the rejection all teach setting a valid bit to invalid for instructions following a taken branch instruction. There is no teaching in Yung that differing instructions are valid or invalid for differing iterations of a loop. InstantWeb merely teaches the existence of software loops and adds nothing to Yung to make obvious differing actions in different loops. Accordingly, claim 7 is not made obvious by the combination of Yung and InstantWeb.

Claim 17 recites subject matter not made obvious by the combination of Yung and InstantWeb. Claim 17 recites "wherein the annul code is loaded into the annul word memory from a selected one of the first data register and the second data register in response to an instruction having a condition predicate." Yung fails to teach this conditional loading of an annul code from one of two registers dependent upon the condition. Yung states at column 6, lines 51 to 53:

> "Alternately, rather than a FIFO, other data storage structures could be used for storing the instruction validity data."

This disclosure proposes an alternate structure for FIFO 66 which stores the validity bits corresponding to the instructions. Claim 17 recites storing alternative annul codes in two registers, then loading the annul word memory with data from a selected register

upon evaluation of the condition predicate. No portion of Yung teaches loading FIFO 66 from a memory or from a register. Instead Yung teaches changing valid bits in FIFO 66 via instruction FIFO control logic 71 under the control of branch unit 18. This invalidation is not disclosed as resulting from loading from a register as recited in claim 17. InstantWeb only teaches the existence of data registers and adds nothing to Yung to make obvious the particular claimed use of two registers. Accordingly, claim 17 is unobvious over the combination of Yung and InstantWeb.

Claim 20 recites subject matter not made obvious by the combination of Yung and InstantWeb. Claim 20 recites "the annul code is loaded into the annul word memory from the register in response to an instruction having a condition predicate." Yung fails to teach this conditional loading of an annul code from a register dependent upon the condition. Yung teaches at column 6, lines 51 to 53 an alternate structure for FIFO 66 which stores the validity bits corresponding to the instructions. Claim 20 recites storing an annul code in a register, then conditionally loading the annul word memory with data from a register upon evaluation of the condition predicate. No portion of Yung teaches loading FIFO 66 from a memory or from a register. Instead Yung teaches changing valid bits in FIFO 66 via instruction FIFO control logic 71 under the control of branch unit 18. This invalidation is not disclosed as resulting from loading from a register as recited in claim 20. InstantWeb only teaches the existence of data registers and adds nothing to Yung to make obvious the particular claimed use of two registers. Accordingly, claim 17 is unobvious over the combination of Yung and InstantWeb.

Claim 21 recites subject matter not made obvious by the combination of Yung and InstantWeb. Claim 21 recites "storing a portion of the annul code into the annul word memory in response to receipt of an interrupt." Yung fails to teach this conditional

loading of an annul code from a register dependent upon an interrupt. Yung teaches at column 6, lines 51 to 53 an alternate structure for FIFO 66 which stores the validity bits corresponding to the instructions. Claim 21 recites conditionally loading the annul word memory with data upon an interrupt. No portion of Yung teaches loading FIFO 66. Instead Yung teaches changing valid bits in FIFO 66 via instruction FIFO control logic 71 under the control of branch unit 18. This invalidation is not disclosed as resulting from loading upon an interrupt as recited in claim 21. Yung includes no teaching of an interrupt. InstantWeb only teaches the existence interrupts and adds nothing to Yung to make obvious the particular claimed use of an interrupt. Accordingly, claim 21 is unobvious over the combination of Yung and InstantWeb.

Claim 18 was rejected under 35 U.S.C. 103(a) as made obvious by the combination of Yung U. S. Patent Number 5,809,324 and Brender et al U.S. Patent Number 3,638,195.

Claim 18 recites subject matter not made obvious by the combination of Yung and Brender. Claim 18 recites "the annul code is loaded into the annul word memory from a selected half of the data register in response to an instruction having a condition predicate." Yung fails to teach this conditional loading of an annul code from a selected half of a register dependent upon an interrupt. Yung teaches at column 6, lines 51 to 53 an alternate structure for FIFO 66 which stores the validity bits corresponding to the instructions. Claim 18 recites conditionally loading the annul word memory with data from a selected half of a register based upon a condition predicate. No portion of Yung teaches loading FIFO 66. Instead Yung teaches changing valid bits in FIFO 66 via instruction FIFO control logic 71 under the control of branch unit 18. This invalidation is not disclosed as resulting from loading upon an interrupt as recited in claim 18. Yung includes no teaching of loading FIFO 66 from a register. Brender

et al only teaches the existence storing data in independent halves of a data register. This disclosure of Brender et al adds nothing to Yung to make obvious the particular claimed use of data stored in halves of a data register. Accordingly, claim 18 is unobvious over the combination of Yung and Brender et al.

The Applicants respectfully request entry and consideration of this amendment. Entry of this amendment is proper at this time because the FINAL REJECTION included new bases for rejection based on newly cited references. Thus the Applicant has had no previous opportunity to respond to these rejections.

The Applicant respectfully submits that all the present claims are allowable for the reasons set forth above. Therefore early entry of this amendment, reconsideration and advance to issue are respectfully requested.

If the Examiner has any questions or other correspondence regarding this application, Applicant requests that the Examiner contact Applicant's attorney at the below listed telephone number and address to facilitate prosecution.

Texas Instruments Incorporated
P.O. Box 655474 M/S 3999
Dallas, Texas 75265
(972) 917-5290
Fax: (972) 917-4418

Respectfully submitted,

Robert D. Marshall, Jr.
Reg. No. 28,527